REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
6 SEP 06	FINAL REPORT	15 MAY 00 TO 30 SEP 05

4. TITLE AND SUBTITLE
CONTROL AND OPTIMIZATION TOOLS FOR SYSTEMS GOVERNED BY NONLINEAR PARTIAL DIFFERENTIAL EQUATIONS

5a. CONTRACT NUMBER
F49620-00-1-0299

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
PROF JEFF BORGGAARD

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
VIRGINIA POLYTECHNIC INSTITUTE & STATE UNIVERSITY
CENTER FOR OPTIMAL DESIGN AND CONTROL
BLACKSBURG, VA 24061

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NL
875 NORTH RANDOLPH STREET
SUITE 325, ROOM 3112
ARLINGTON, VA 22203-1768

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT

AFRL-SR-AR-TR-06-0395

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVE FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We have developed a number of theoretical and computational tools for optimal design and control of spatially distributed systems. Our main results were focused on complex fluid systems modeled by the Navier-Stokes equations. We considered turbulent flows, thermal fluids, temperature dependent material properties and time dependence among other complexities. Sensitivity analysis, the process of quantifying the dependence of parameters on these flows, was performed for a number of interesting flow problems. We investigated methods for computing sensitivity variables including a novel application of automatic differentiation (AD) technology as well as the implementation of a solver for a general sensitivity equation. This solver includes adaptive mesh refinement for the coupled flow and sensitivity equations. Our research on advanced computational fluid dynamics (CFD) simulation and sensitivity analysis continues, with the development of a parallel 3D finite element based software package to take advantage of modern cluster

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*

CONTROL AND OPTIMIZATION TOOLS FOR SYSTEMS GOVERNED BY NONLINEAR PARTIAL DIFFERENTIAL EQUATIONS

AFOSR F49620-00-1-0299

FINAL REPORT

Jeff Borggaard
Center for Optimal Design and Control
Interdisciplinary Center for Applied Mathematics
Virginia Tech

OBJECTIVES

The primary objectives of this research was to develop innovative multi-functional computational tools for optimal design and control of spatially distributed systems.

ACCOMPLISHMENTS

Overview

We have developed a number of theoretical and computational tools for optimal design and control of spatially distributed systems. Our main results were focused on complex fluid systems modeled by the Navier-Stokes equations. We considered turbulent flows, thermal fluids, temperature dependent material properties and time dependence among other complexities. Sensitivity analysis, the process of quantifying the dependence of parameters on these flows, was performed for a number of interesting flow problems. We investigated methods for computing sensitivity variables including a novel application of automatic differentiation (AD) technology as well as the implementation of a solver for a general sensitivity equation. This solver includes adaptive mesh refinement for the coupled flow and sensitivity equations. Our research on advanced computational fluid dynamics (CFD) simulation and sensitivity analysis continues, with the development of a parallel 3D finite element based software package to take advantage of modern cluster-based computer architectures.

In addition to our improved simulation and sensitivity analysis capability, we considered three main application areas: control, optimization, and novel uses for sensitivity analysis (including uncertainty quantification). We provide more detailed descriptions of our research in the sections below.

High Performance CFD Simulation

The two most popular methodologies for simulation of turbulent flows in engineering problems are large-eddy simulation (LES) and Reynolds-averaged Navier-Stokes (RANS). Both methods reduce computational requirements by computing averaged flow quantities over space (LES) or time (RANS). However, these approaches present the closure problem, the

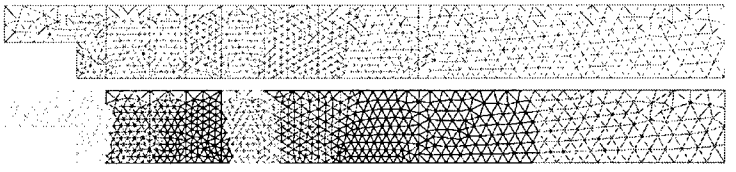


Figure 1: Initial Mesh and Optimal Partition

need to model fluctuations by the averaged quantities. The influence of these, sometimes empirical, models on the CFD results are important to quantify when interpreting solutions.

We have used both methodologies in our research. In the remainder of this section, we will outline our LES flow solver, present a novel implementation of time varying boundary conditions (required for simulating injection-based flow control) and a sensitivity analysis of turbulent flow with respect to the most prevalent LES closure model known as the Smagorinsky model. This sensitivity analysis was carried out using AD.

Our work with RANS will be discussed in the next section where it is used as a basis to present our work on general sensitivity equation solvers.

Parallel Implementation of LES Models

LES is a natural tool for studying flow control. It has the ability to model complex flows in complex domains. It is not possible to use, so-called, direct numerical simulation to simulate these flows. To make this method amenable to flow control, and flow simulation in general, a number of issues associated with model closure and boundary conditions need to be addressed. The latter issue is critical since most flow control is effected with actuation on the boundary.

We have developed ViTLES (the Virginia Tech Large-Eddy Simulator) to study these issues and eventually simulate controlled flow. This software is designed to run on *System X*, the unique terascale computing facility built at Virginia Tech. ViTLES combines finite element methods, implicit time-stepping and parallel linear algebra to carry out flow simulations. Near optimal load balancing is carried out by mesh partitioning (see Figure 1) based on the Metis graph partitioning library.

New Models in Large-Eddy Simulation

LES is carried out by convolving the Navier-Stokes equations with a spatial filter (usually the Gaussian averaging filter) and averaging radius δ. This averaging operation is usually denoted with an overbar $\bar{z} \equiv g_\delta \star z$. The averaged equations leads to a model that estimates fluid dynamics in spatial regions. Since averaged quantities are smoother, they are easier to compute. This averaging, however, leads to a number of mathematical complications that affect the accuracy of the simulations. The most frequently studied is the *closure problem* where the term such as $\bar{z}\bar{z}$ needs to be modeled using the quantity $\bar{z}$ (note that $\bar{z}\bar{z} \neq \bar{z}\bar{z}$). Although many important models are being developed and tested, most codes use the Smagorinsky model which approximates the new term

$$\tau(\bar{z}) \approx (c_s\delta)^2 \left|\nabla\bar{z}\right| \nabla\bar{z}.$$

It is important to understand the effect of the filter width (averaging radius) on the flow. To investigate this, we present the flow past a backward-facing step (depicted in Figure 1)

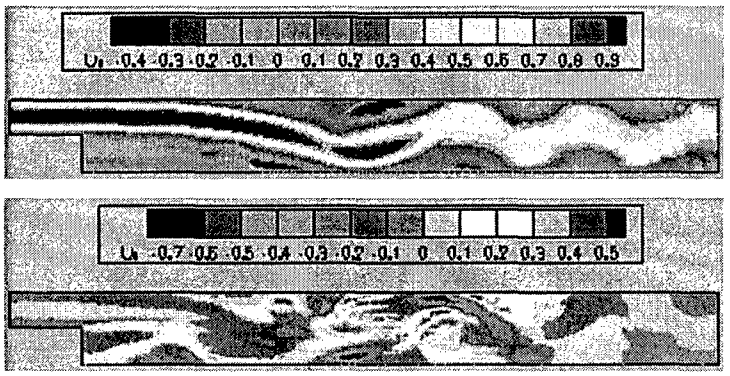


Figure 2: Horizontal Velocity (top) and its Sensitivity with respect to δ (bottom)

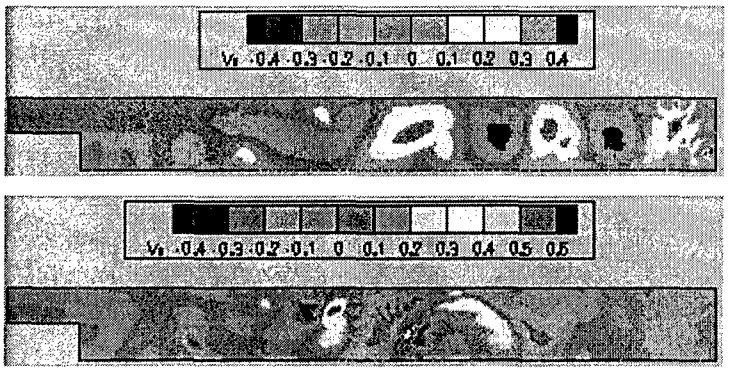


Figure 3: Vertical Velocity (top) and its Sensitivity with respect to δ (bottom)

at a Reynolds number of 1000, large enough to lead to a turbulent flow. A time snapshot of the flow is depicted with the horizontal (Figure 2) and vertical (Figure 3) components of velocity and the pressure (Figure 4). Along with these figures, we also show the sensitivity of the flow with respect to δ. This is small in many portions of the domain, indicating that the closure model may not have much influence in these regions. However, note that the sensitivity is large in regions with high vorticity and contains smaller structures than the flow itself. This corresponds to the fact that an extrapolation of the flow to $\delta = 0$ would lead to fully resolved turbulent flow (without spatial averaging). For this reason, it is much more difficult to obtain the sensitivity than the flow in turbulent problems.

Typically, these models allow the filter width (averaging radius) to go to zero near walls to handle boundary conditions. However, this then requires fine discretizations to accurately resolve the flow in these regions. This negates the utility of LES as a low-order model. We are looking at LES models which allow for constant (large) filter widths and while only capturing large structures in the flow, can do this with a coarse discretization.

The important application is the natural extension of large-eddy simulation to flows with time-varying boundary conditions (as is the case with flow control based on injection). There are two approaches used to handle boundary conditions. The first is near wall resolution. This approach requires fine meshes near the boundary and may be computationally expensive. The second approach is near wall modeling. This approach uses boundary

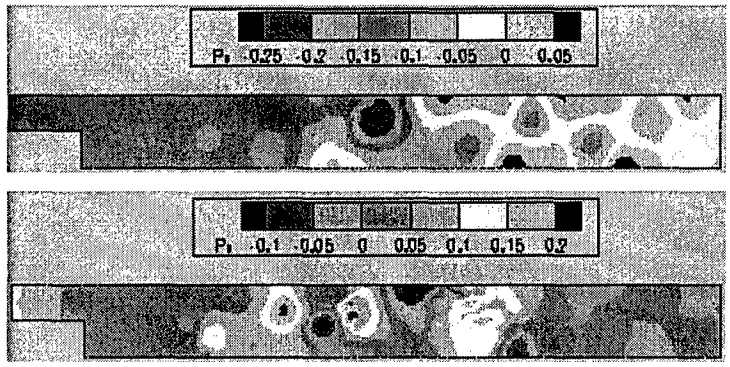


Figure 4: Pressure (top) and its Sensitivity with respect to δ (bottom)

layer theory to generate wall functions and avoids fine meshes. However, the usual wall models do not cover time varying boundary conditions. We have introduced a mathematical solution based on approximate deconvolution and explicitly accounting for the boundary commutation error term. Boundary conditions are ultimately determined by solving small elliptic problems along the boundary of the domain.

Sensitivity Analysis for Complex Flows

Sensitivity of Turbulence Models

A standard model of turbulent flow utilizes the k–ϵ turbulence model for the turbulence kinetic energy (k) and its dissipation rate (ϵ) to determine the eddy viscosity

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon}.$$

This model is used to close the Reynolds-averaged Navier-Stokes equations,

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \rho \mathbf{u} \cdot \nabla \mathbf{u} &= -\nabla p + \nabla \cdot \left[(\mu + \mu_t) \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \right] + \mathbf{f}. \end{aligned}$$

The transport equations for k and ϵ suggested by Launder and Spalding are given as

$$\rho \mathbf{u} \cdot \nabla k = \nabla \cdot \left[\left(\mu + \frac{\mu_t}{\sigma_k} \right) \nabla k \right] + \mu_t \nabla \mathbf{u} : \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) - \rho^2 C_\mu \frac{k^2}{\mu_t} + q_k$$

and

$$\rho \mathbf{u} \cdot \nabla \epsilon = \nabla \cdot \left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon} \right) \nabla \epsilon \right] + \rho C_1 C_\mu k \nabla \mathbf{u} : \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) - C_2 \rho \frac{\epsilon^2}{k} + q_\epsilon.$$

The constants C_1, C_2, C_μ, σ_k, and σ_ϵ are given in Table 1.

For accuracy and stability considerations, we perform a change of variables in the above equations to solve for the logarithms of k and ϵ. This assures positivity and that the variables have smoother variations. The standard k–ϵ turbulence model is not valid when the

C_μ	C_1	C_2	σ_k	σ_ϵ
0.09	1.44	1.92	1.0	1.3

Table 1: Constants for the k–ϵ model

turbulence Reynolds number is low. This happens near the wall and is treated using wall functions to model the flow in this region.

The continuous sensitivity equations (CSE) are derived formally by implicit differentiation of the flow equations with respect to a parameter a. Thus, not only do we treat the variable $\mathbf{u}$ as a function of space, but also as a function of the parameter a. The key point here is that we adopt a *general* approach: we consider *any* (non-geometric) parameter a. Consequently, all the quantities involved (flow variables (u,p), material properties (e.g. ρ, μ, k), coefficients (e.g. C_μ, C_1),...) may simultaneously depend on a. When specific parameters are selected, certain terms may simply vanish from the general equation. The user must specify the fluid properties (ρ, μ, etc.) for the flow along with their sensitivities (ρ', μ', etc.) for the sensitivity equation.

As in our previous work, we are careful to perform mesh adaptation using all solved quantities ($\mathbf{u}$, k, ϵ, and their sensitivities). The ability to tailor software to calculate accurate sensitivity information is one of the main advantages to our continuous approach.

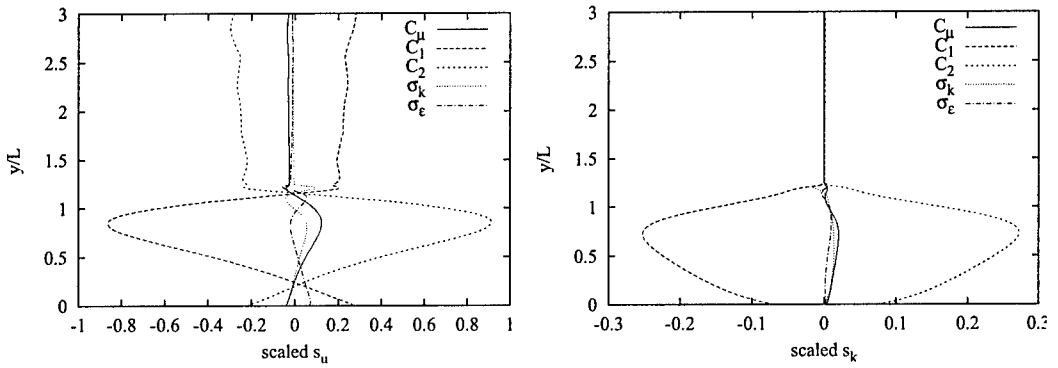
To demonstrate the flexibility of this formulation, we calculate sensitivities for two example problems. The first is flow over a backward facing step at a Reynolds number of 47,625 (based on step height $L = 1$) with constant Dirichlet boundary conditions at the inflow ($2L$ wide) of $u = 1$, $v = 0$, $k = 0.005$ and $\epsilon = 0.01$. An adapted finite element mesh of 60,000 nodes was used to carry out the approximation.

In Figure 5, we plot the sensitivity of the velocity to all of the modeling coefficients appearing in Table 1. The sensitivities are scaled by the nominal values of the coefficients. Observe that the flow along these lines would not be significantly affected by the same relative increases in both C_1 and C_2 and likewise not affected by increases in both σ_k and σ_ϵ. This suggests that these pairs of coefficients may not be independent for this flow. In fact, we've observed similar behavior in many areas of the flow and in the skin friction sensitivity on the wall downstream of the step.

The second example is the development of a turbulent boundary layer by flow impinging on a flat plate at a Reynolds number of 2×10^6. In Figure 6, we present the mesh used for this calculation along with contours of the horizontal velocity component along with the sensitivity of the (logarithm of the) turbulent kinetic energy with respect to the coefficient C_1. By observing these two contours, we can explain the pattern in the adaptive mesh refinement.

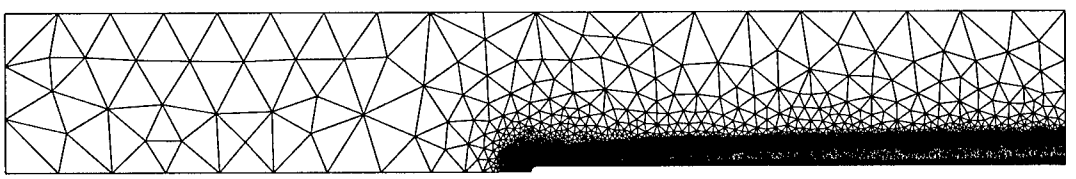
Applications of Sensitivity Analysis

Sensitivity analysis is an important tool in the control and optimization of fluid systems. It also has a number of other uses, including uncertainty quantification, estimating nearby

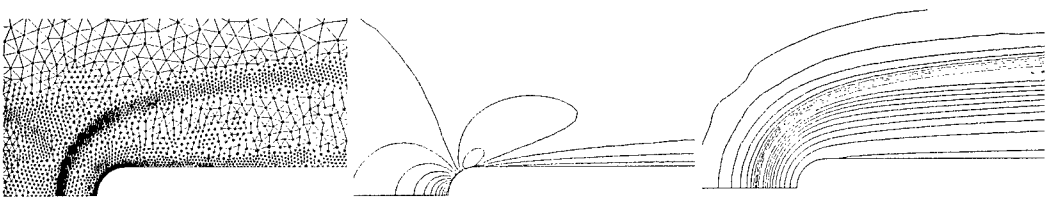


Horizontal Velocity Sensitivity Turbulent Kinetic Energy Sensitivity

Figure 5: Scaled sensitivities at $x = 8L/3$



Final adapted mesh



Zoom of the final mesh u $\frac{\partial K}{\partial C_1}$

Figure 6: Flat plate: final mesh and solution

flows, and determining the relative importance of model parameters (as illustrated by the scaled sensitivities computed above). We will discuss their role in computing gradients for optimal design and uncertainty quantification below. The use of sensitivity analysis in answering actuator placement questions is discussed in the next section on Linear Feedback Control for Distributed Parameter Systems. Other applications were described in annual reports.

Optimal Design

In this section, we demonstrate our use of sensitivity analysis for optimization of a thermal/fluid system. Convection is often used as a mechanism for cooling. Consider flow in the domain depicted in Figure 7. In this problem, fluid at a relatively cool ambient temperature is injected at the inflow for the purpose of convecting heat away from the block on the side of the wall. The fluid flow in this problem can be modeled by the equations of mixed convection; the Navier-Stokes (with buoyancy term), continuity, and energy equa-

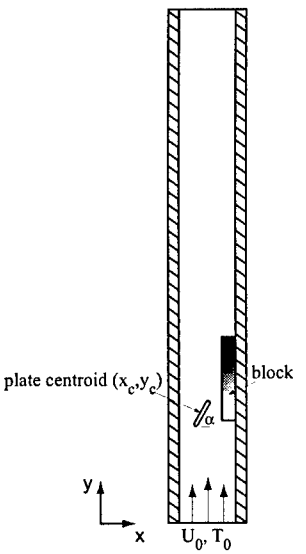


Figure 7: Mixed Convection Flow Problem

tions along with the appropriate boundary conditions:

$$\begin{aligned} \rho\mathbf{u}\cdot\nabla\mathbf{u} &= -\nabla p + \nabla\cdot\tau(\mathbf{u}) - \rho\mathbf{g}\beta\left(T-T_0\right) + \mathbf{f}, \\ \nabla\cdot\mathbf{u} &= 0, \\ \rho c_p\mathbf{u}\cdot\nabla T &= \nabla\cdot(\kappa\nabla T) + q \end{aligned}$$

where the viscous fluid stress is given by

$$\tau(\mathbf{u}) = \mu\left\{\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right\}.$$

Thus, we consider flows for which the buoyancy term $\rho\mathbf{g}\beta\left(T-T_0\right)$ is significant. At the inflow, we assume that the flow velocity is constant, $U_0\hat{\mathbf{j}}$, with an ambient temperature, $T_0 = 0$. The walls of the channel are assumed insulated and the block maintains a uniform temperature of $T_b = 1$. At the outflow, we specify a zero stress condition.

The flow domain also contains a plate of non-dimensional length 0.25 based on the channel width or block length (with parabolic ends extending 0.01 on each side of the plate with a width of 0.02). The thickness of the block is 0.2 and begins one channel width above the inflow. The plate is used to enhance the heat transfer properties of the system, directing more of the cooler flow towards the block and creating a thinner thermal boundary layer. The temperature in this plate can be described by the Laplace equation. While the velocity of the fluid vanishes on the plate surface, Γ_a, the temperature in the plate is coupled to the temperature in the flow since the temperature and heat flux are continuous at the interface. Thus, the interface condition is

$$\begin{aligned} T_{\text{fluid}} &= T_{\text{plate}} \qquad \text{and} \\ \kappa_{\text{fluid}}\nabla T_{\text{fluid}} &= \kappa_{\text{plate}}\nabla T_{\text{plate}} \quad \text{on } \Gamma_a. \end{aligned}$$

For this problem, we assume that the conductivity of the fluid and the plate are the same, or $\kappa(x,y) = \kappa$.

Given this system, a natural design problem is to specify the location of the plate, describing the coordinates of the centroid (x_c, y_c) and the angle α, to achieve the maximum cooling benefit from this strategy. To describe this problem precisely, we need to quantify the effectiveness of a given design. For a given vector of design parameters $\mathbf{a} = (x_c, y_c, \alpha)$ describing the plate geometry (and hence the flow domain), let $u(\cdot; \mathbf{a})$, $v(\cdot; \mathbf{a})$, $p(\cdot; \mathbf{a})$ and $T(\cdot; \mathbf{a})$ be the solution to the flow equations. Then a measure of the heat transfered off the block in this configuration is

$$\mathcal{J}_1(\mathbf{a}) = \int_{\Gamma_b} \kappa \nabla T(x, y; \mathbf{a}) \cdot \hat{n} \, \mathrm{d}\Gamma_b. \tag{1}$$

Mathematically, the optimal design problem above can be viewed as finding the maximum of a function that depends on the design parameters through the solution of the flow equations. In practice, this solution has to be found using numerical techniques. We use an adaptive finite element method as discussed below. Once the approximation is defined, a straight-forward technique for finding the optimal parameter values is to couple the approximate objective function with an optimization algorithm. Since every function evaluation requires the approximation of a complicated flow field, optimization algorithms that reduce the number of function evaluations are desirable. Gradient-based algorithms are typically used since gradients can often be found for a fraction of the cost of performing a flow calculation. This usually involves solving either an adjoint equation or a number of sensitivity equations. Since we have a relatively small number of design parameters (3 or less) for this problem, we consider using sensitivity equations. The details of this methodology are provided below.

Since the amount of time required to evaluate the function dwarves the time required to estimate the next step, it makes sense to consider algorithms which try to find the best possible step. This leads us to consider gradient-based optimization methods since, as we describe below, we can simultaneously find the gradient for a fraction of the cost of performing a nonlinear flow solve. To minimize the number of iterations required, we look at higher order methods such as quasi-Newton methods. We avoid calculating the Hessian, $\nabla^2 \mathcal{J}_1(\cdot)$, by using a BFGS secant update strategy and initialize our approximate Hessian with $H_0 = \mathcal{J}_1(\mathbf{a}_0) I$. This gives up some of the accelerated convergence in Newton's method for the advantage of avoiding the computational difficulties of computing the true Hessian at every step. A thorough study weighing the advantages/disadvantages of calculating the Hessian vs. using a secant update (like BFGS) needs to be performed for optimization problems of this type.

To allow convergence for a wider range of initial parameter values, we consider using a trust-region globalization strategy. Thus, at a current design point $\mathbf{a}_k$, we choose the next parameter value which satisfies the following trust-region sub-problem:

$$\max_{\|\mathbf{s}_k\| < \delta_k} \mathcal{J}_1(\mathbf{a}_k) + \nabla \mathcal{J}_1(\mathbf{a}_k)^T \mathbf{s}_k + \frac{1}{2} \mathbf{s}_k^T H_k \mathbf{s}_k,$$

then $\mathbf{a}_{k+1} = \mathbf{a}_k + \mathbf{s}_k$ if $\mathcal{J}_1(\mathbf{a}_i)$ is determined to be a satisfactory point. This trust-region strategy essentially expands the radius of convergence of quasi-Newton methods and has other properties that make it attractive for solving optimal design problems in this framework.

Most optimization algorithms implement an approximation to the trust-region subproblem above, however, for our situation, we can easily afford to solve this subproblem precisely.

Gradient Calculations: The gradient of the objective functions can be derived by implicit differentiation (recall that we consider our flow field to be a function of our design parameter a) leading to

$$\frac{\partial}{\partial a_i}\mathcal{J}_1(\mathbf{a}) = \int_{\Gamma_b} \kappa \nabla s_T(x, y; \mathbf{a}) \cdot \hat{n} \, \mathrm{d}\Gamma_b.$$

where $s_T = \frac{\partial}{\partial a_i} T$.

This sensitivity variable, along with $s_u = \frac{\partial}{\partial a_i} u$ and $s_p = \frac{\partial}{\partial a_i} p$ satisfy the continuous sensitivity equation which can be derived formally by implicitly differentiating the Navier-Stokes equations and boundary conditions with respect to the parameter a_i:

$$\begin{aligned} \rho\left(\mathbf{s}_u \cdot \nabla \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{s}_u\right) &= -\nabla s_p + \nabla \cdot \tau(\mathbf{s}_u) - \rho \mathbf{g} \beta s_T + \mathbf{f}_s \\ \nabla \cdot \mathbf{s}_u &= 0 \\ \rho c_p\left(\mathbf{s}_u \cdot \nabla T + \mathbf{u} \cdot \nabla s_T\right) &= \nabla \cdot (\kappa \nabla s_T) + q_s \end{aligned}$$

where we have assumed, among other things, that ρ, $\tau(\cdot)$, c_p and κ are independent of the design parameter. If this were not the case, we would have additional source terms in the sensitivity equations above.

In order to complete the description of the partial differential equation, we need to differentiate the boundary and interface conditions with respect to $\mathbf{a}$. This is straight-forward for all of the fluid boundaries except for the plate. In this case, we need to set the total derivative equal to zero and solve for the sensitivity boundary conditions. For example, since we have the condition $u = 0$ on the plate, if we differentiate that condition with respect to the parameter x_c, we find

$$s_u = -\frac{\partial u}{\partial x} \qquad \text{on } \Gamma_a.$$

Other conditions are derived similarly.

Consider flow in our optimal design problem for the case where $x_c = 0.55$, $y_c = 1$, $\alpha = 60°$, $Re = 378$, $Pr = 0.7$ and $Ri = 1.0$. Using our adaptive finite element method, we compute the flow and corresponding sensitivity with respect to the parameter α.

In Table 2, we plot the value of the function and it's derivative using each expression given in the problem description. We see that as the mesh is refined, the values of these functions converge. Both of these objective functions converve to the same value, indicating that the energy is conserved. Although this convergence indicates that we aren't losing much information downstream, the change in $\mathcal{J}_1$ over the last cycle was less than that of $\mathcal{J}_2$. Therefore, we decided to use $\mathcal{J}_1$ to perform our optimization, viewing this as a more accurate measure. The disadvantage of using this choice of the objective function is that it requires the derivative of the sensitivity on the block surface to compute the gradient.

This table gives us important information about how to choose stopping criteria for our optimization algorithm. For instance, since we are going to perform optimization using $\mathcal{J}_1$,

we see that we can't expect to find the optimum value of $\mathcal{J}_1$ to any tolerance smaller than 1×10^{-4} or expect the gradient to be accurate to any tolerance smaller than 1×10^{-4} either.

A typical mesh is shown in Figure 8. This indicates that mesh adaptation is performed in the regions where the flow physics vary the strongest–both at the ends of the plate and along the thermal boundary layer and corners of the block. Not only is this adaptation method attempting to provide the most accurate flow and sensitivity solution for given computer resources, but is generating this graded mesh in a rather automatic fashion to minimize the human interaction at each design iteration. We discuss the details of the optimization below.

Nodes	$\mathcal{J}_1$	$\mathcal{J}_2$	$\nabla\mathcal{J}_1$	$\nabla\mathcal{J}_2$
736	0.06274	0.06673	-0.00723	-0.00996
1346	0.06469	0.06410	0.00006	-0.00033
2775	0.06599	0.06598	-0.00790	-0.00770
5772	0.06741	0.06742	-0.01304	-0.01305
13505	0.06837	0.06835	-0.01374	-0.01367
32476	0.06891	0.06898	-0.01442	-0.01443

Table 2: Function and Gradient Values at 60 deg

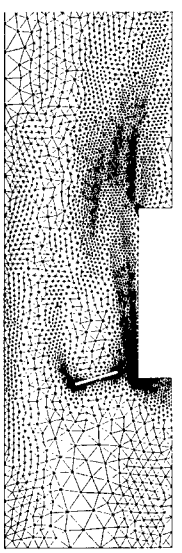


Figure 8: Close-up of a Typical Mesh

As an initial optimization problem, we consider the case where the centroid of the plate is fixed, but the optimum angle needs to be determined. For this one parameter problem, we plot a few values of $\mathcal{J}_1$ just to get an idea what the objective function looks like. The flow approximations are determined after 5 adaptive cycles resulting in approximately 30,000 nodes. The resulting function is plotted in Figure 9. Note that while the objective function

is clearly not concave, it does seem concave in the region near the optimum. Furthermore, the optimum value seems to occur between the value of 15 and 20 degrees.

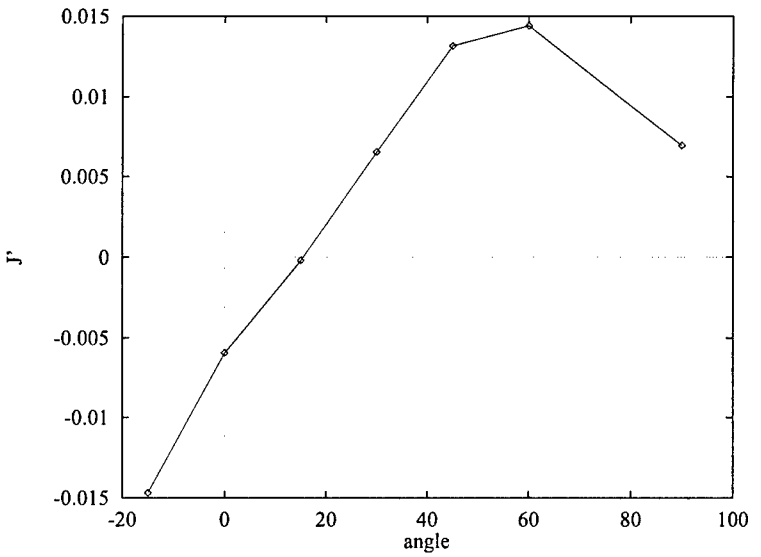


Figure 9: Design Objective Function, $\mathcal{J}_1$

Based on the validation example and the adaptation history of $\mathcal{J}_1$ and $\nabla\mathcal{J}_1$ for 60° (corresponding to a more difficult flow and sensitivity field), we estimate that we have less than 1×10^{-4} accuracy in both the function and the gradient. We can use this information to select stopping criteria for the optimization algorithm. Starting from 60°, we run our optimization algorithm with an initial trust-region radius of 20°. The iteration history is provided in Table 3. We see that we run up against the trust-region radius for the first two iterations (at this time, the Hessian is being updated with information). The final step looks for a value of $\mathcal{J}_1$ at 15.917° and finds a function value which decreases slightly, but outside our confidence value, and a gradient which is near our confidence value. At this point, we halted the iteration as the next point was being sought at the halfway point between 15.917 and 20. The result is about a 10 percent increase in performance of the system. The gain over the flow configuration with no plate is nearly 25 percent ($\mathcal{J}_1 = 0.06116$).

Iter.	α	$\mathcal{J}_1$	$\nabla\mathcal{J}_1$
0	60.000	0.0689099	-0.0144212
1	40.000	0.0736174	-0.0112043
2	20.000	0.0757117	-0.0019027
3	15.917	0.0755710	-0.0003157

Table 3: One Parameter Iteration History

We repeat the design problem at the same flow conditions as above, however, we now seek the centroid of the plate as well as the angle. We start from the same location as the one parameter optimization problem. Since we expected more function evaluations in this

example, we used a coarse mesh for the first six iterations in order to get a good initial guess more cheaply. Beginning with the seventh iteration, we use five mesh adaptation cycles at about 30,000 nodes to perform the flow and sensitivity calculations. The result is improved performance over the case where the centroid is fixed (as expected). The performance increases about a 13 percent by moving the plate closer to the center of the block. The iteration history is provided in Table 4.

Iter.	x_c	y_c	α	$\mathcal{J}_1$	$\|\nabla\mathcal{J}_1\|$
0	0.550	1.000	60.00	0.06794	0.0297
1	0.458	1.263	53.68	0.06850	0.0192
2	0.504	1.132	56.84	0.06884	0.0194
3	0.453	1.166	51.33	0.06911	0.0182
4	0.444	1.193	47.20	0.07022	0.0180
5	0.441	1.231	38.90	0.07209	0.0149
6	0.426	1.274	30.71	0.07235	0.0241
7	0.443	1.325	22.97	0.07660	0.0129
8	0.429	1.338	6.46	0.07870	0.0086
9	0.413	1.331	-13.88	0.07842	0.0056
10	0.425	1.313	-3.62	0.07887	0.0043
11	0.433	1.270	-2.80	0.07931	0.0022

Table 4: Three Parameter Iteration History

The temperature contours corresponding to no plate in the problem, the plate with an optimum angle, and the plate at the three parameter maximum are provided in Figure 10. Note that the plate has the effect of pinching the boundary layer, thereby causing an increase in heat flux off the block. One can also see the complex temperature profile behind the block, although most of the heat is lost over the blunt edge of the plate and the long surface.

Figure 11 displays the temperature flux as a function of the arc length of the block. Thus, from 0 to 0.2, we find the temperature flux on the leading edge of the block, etc. The integral of this function provides the objection function $\mathcal{J}_1$. By introducing the plate, the temperature flux is increased on the long edge of the block (from $s = 0.2$ to $s = 1.2$), especially towards the inflow. The ability to slide the plate toward the center for the block spreads out this positive temperature flux over the trailing edge of the long edge of the plate. This results in the increased efficiency.

Uncertainty Quantification

One potentially important application of sensitivity analysis is in developing bounds for CFD simulations with uncertain parameters. For example, in turbulence models such as k–ϵ (Launder and Spalding), there are a number of closure coefficients that are determined by fitting experimental results. For prediction, these same coefficients are often applied to different flow regimes than those used to construct them. These are inherently uncertain, and this uncertainty should be propagated to the final flow solution. Using sensitivity

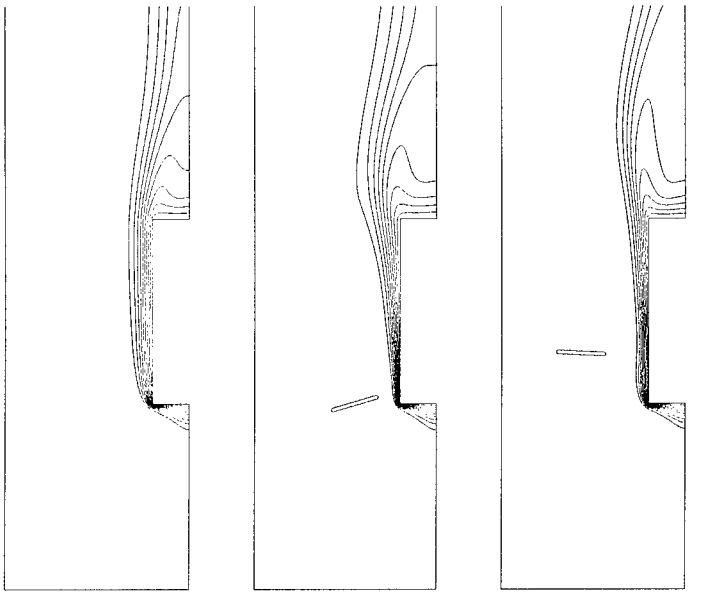


Figure 10: Temperature Contours

analysis, we use the heuristic bound

$$|\Delta u| \approx \sum_{i=1}^{n} \left| \frac{\partial u}{\partial a_i}(x, y; \mathbf{a}) \right| |\Delta a_i|$$

where the parameters a_i represent the five closure coefficients: C_μ, C_1, C_2, σ_k, and σ_ϵ with assumed uncertainty in the last significant digit. We demonstrate the promise of these bounds when we compare our turbulent flow simulation (Re=47,625) over a backward facing step with published experimental measurements [Kim, 78] in Figure 12. Observe that the flow in the center of the channel is relatively insensitive to perturbations in the coefficients where some of the discrepancy in the CFD/experimental comparison near the wall may be explained by these perturbations. A full comparison would also include uncertainty bounds on the experimental data as well.

Linear Feedback Control for Distributed Parameter Systems

Chandrasekhar algorithms

We have developed MATLAB code based on 4th order Runge-Kutta methods for integrating the matrix differential equations backward in time to steady-state. As a test of the effectiveness of this strategy for computing gains, we compared CPU times for gain calculations with those computed using MATLAB's built-in Riccati equation solver `lqr`. In simulations run with Chris Camphouse, we tested this algorithm on a Dirichlet boundary control problem for the two dimensional Burgers equation. Here, we consider "flow" in a channel with control occurring over a submerged obstruction (where control is applied). The control output is the average value of "velocity" over a patch downstream of the obstruction. A

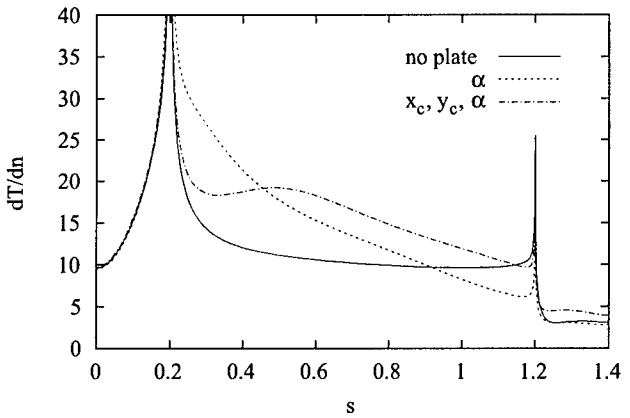


Figure 11: Temperature Flux Along Block

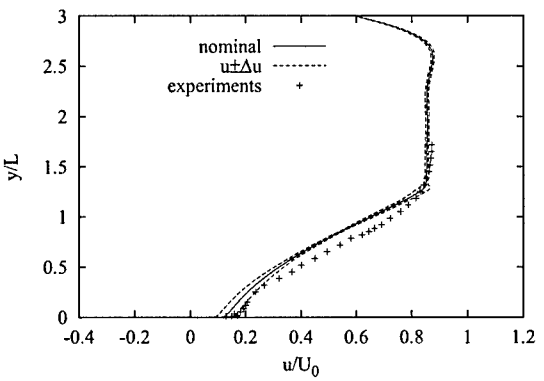


Figure 12: Uncertainty in u at $x/L = 8$

summary of the results for different outflow boundary conditions and two values of the weighting parameter α are reported in Table 5. For the standard LQR problem ($\alpha = 0$), the Chandrasekhar equations are extremely competitive, offering nearly an order of magnitude improvement in the computational time required to compute gains. What makes this more promising is the fact that we are comparing interpreted m-file code to compiled routines in MATLAB. A comparison of more optimized Chandrasekhar code would lead to more savings.

Sensitivity Analysis for Chandrasekhar and Riccati Equations

We have implemented software to solve for quantities $\frac{\partial \Pi}{\partial \alpha}$ and $\frac{\partial h}{\partial \alpha}$, where h is the function gain. In the former case, we utilize standard Lyapunov solvers to compute the desired sensitivity information. In the latter, we integrate a coupled set of Chandrasekhar and sensitivity equations backward in time to steady-state. For some problems, we can use either Π or h to measure the performance of the controlled system. The sensitivity of h with respect to α would then provide an efficient means of calculating gradients for an optimization algorithm.

α	Outflow BC	Riccati	Chandrasekhar
0	Neumann	36.05	4.57
	Dirichlét	32.09	4.49
	Robins	35.52	4.41
	Periodic	34.35	4.36
0.4	Neumann	37.40	67.62
	Dirichlét	30.52	17.18
	Robins	36.29	11.80
	Periodic	34.65	13.89

Table 5: CPU Minutes to Compute Functional Gains

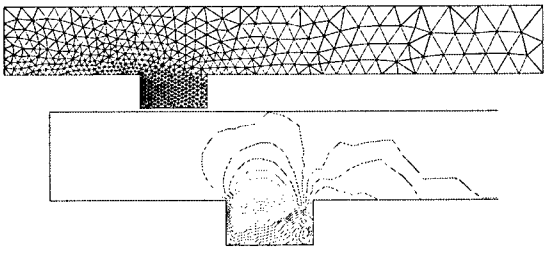


Figure 13: Mesh and Functional Gain Streamlines

Adaptive Mesh Refinement and Conditioning of Riccati Equations Since large Riccati problems are expensive and/or intractable, it is advantageous to use adaptive mesh refinement when computing functional gains. The primary advantage is accurate computation of these gains while keeping the problem size relatively small. As seen in Figure 13, most of the degrees of freedom in computing gains that stabilize flow in a driven cavity would be needed in the region of the cavity. What we have discovered is that consideration of the stability/"solvability" of the resulting Riccati problem is also required.

To highlight this issue, we consider the Dirichlet boundary control problem for the heat equation on the unit interval with control applied on the right hand boundary. For certain controlled outputs, the functional gains can exhibit a singularity (details of the problem are described in Reference 6). A few applications of adaptive mesh refinement cluster elements near the singularity and provide substantial improvement in the computed gains. However, additional adaptivity cycles lead to a breakdown in the gains which can be explained by considering the effect of stretched meshes on solutions to the Riccati equation:

$$\mathbf{A}_N^*\mathbf{\Pi}_N + \mathbf{\Pi}_N\mathbf{A}_N - \mathbf{\Pi}_N\mathbf{B}_N\mathbf{R}^{-1}\mathbf{B}_N^*\mathbf{\Pi}_N + \mathbf{Q}_N = 0. \tag{2}$$

If we consider the Riccati equation (2), then we would like to guarantee the stability of solutions $\mathbf{\Pi_N}$ due to perturbations in $\mathbf{A_N}$, $\mathbf{B_N}$, or $\mathbf{Q_N}$. If, for example, small changes in $\mathbf{A_N}$ lead to large changes in $\mathbf{\Pi_N}$, and ultimately in $\mathbf{K_N}$, then we can expect computational difficulty and misleading solutions. Thus, following the discussion in Datta, we compute bounds on the condition number of the Riccati equations:

$$L \leq \kappa_\Pi \leq U. \tag{3}$$

Table 6: Uniform mesh: Bounds on Riccati Condition Number

Refinement Cycle	Elements	Residual of (2)	L	U	$\kappa(A)$
0	25	5.34×10^{-10}	2.732×10^{5}	2.740×10^{5}	5.23×10^{2}
1	50	2.43×10^{-9}	2.499×10^{6}	2.502×10^{6}	2.11×10^{3}
2	100	1.45×10^{-8}	2.374×10^{7}	2.375×10^{7}	8.48×10^{3}
3	200	1.46×10^{-7}	1.952×10^{8}	1.953×10^{8}	3.40×10^{4}
4	400	1.75×10^{-6}	1.583×10^{9}	1.584×10^{9}	1.36×10^{5}

Table 7: Adapted mesh: Bounds on Riccati Condition Number

Adaptation Cycle	Elements	Residual of (2)	L	U	$\kappa(A)$
0	20	2.80×10^{-13}	1.095×10^{5}	1.100×10^{5}	3.32×10^{2}
1	36	6.12×10^{-13}	2.589×10^{8}	2.589×10^{8}	1.54×10^{5}
2	45	3.48×10^{-9}	1.286×10^{12}	1.286×10^{12}	9.47×10^{6}
3	48	1.16×10^{-1}	5.261×10^{15}	5.261×10^{15}	9.52×10^{7}
4	53	2.17×10^{-2}	1.884×10^{16}	1.884×10^{16}	4.05×10^{9}
5	45	7.24×10^{0}	2.776×10^{19}	2.776×10^{19}	1.19×10^{9}
6	41	2.56×10^{2}	3.469×10^{16}	3.469×10^{16}	2.48×10^{10}
7	46	5.24×10^{-2}	3.741×10^{15}	3.741×10^{15}	2.38×10^{10}

This is analogous to the stability problems caused by large condition numbers in the solution of linear systems (cf. Golub and Van Loan). In other words, κ_Π gives us a relationship between relative changes in the components of (2) and relative changes in the Riccati solution,

$$\frac{\|\Delta\Pi\|}{\|\Pi\|} \leq \kappa_\Pi \left\{ \frac{\|\Delta Q\|}{\|Q\|} + \frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta(BR^{-1}B^T)\|}{\|BR^{-1}B^T\|} \right\}.$$

We present bounds on the condition number using either uniform meshes (Table 6) or adapted meshes (Table 7). We see that the bounds on errors in the condition number grow as the mesh is refined in both cases. However, when the meshes are heavily stretched through adaptivity, the solution breaks down after four mesh refinement cycles. Therefore, research into approximation schemes leading to stable Riccati problems is needed for problems with fine scale structures.

Optimal Actuator Placement

We have constructed a number of design objectives to be used for actuator placement. These extend the usual LQR-type control cost used in many optimal placement strategies by considering spatially distributed disturbance functions. We outline these design objectives using a simple one-dimensional heat equation model.

We consider the problem of placing a heat source/sink for control of temperature z in a

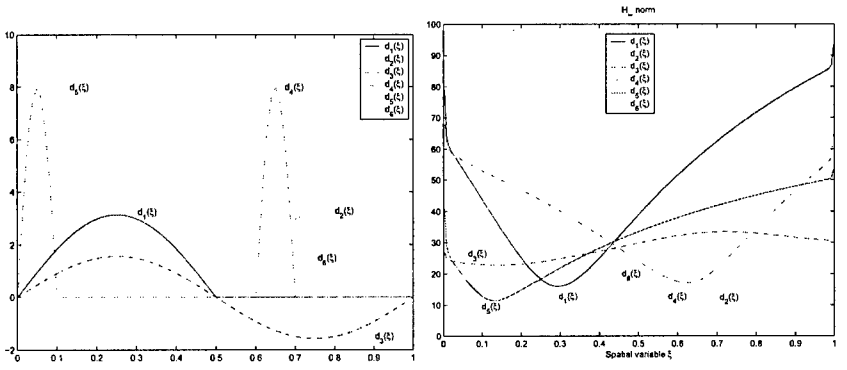


Figure 14: Distributed Disturbances and Design Objective

rod modeled by the heat equation,

$$z_t(t,x) = z_{xx}(t,x) + b(x;\alpha)u(t) \quad \text{with} \quad z(t,0) = 0 = z(t,1),\ t > 0$$

and initial conditions $z(0,x) = z_0(x)$. Note that the location of the source is parameterized by α. We consider the introduction of a distributed disturbance function of the form $d(x)w(t)$. The standard optimal LQR-cost design problem is to find the value of α_* that minimizes the LQR cost over a set of possible initial data $\bar{Z}$. *Minimize*

$$\max_{z_0 \in \bar{Z}} J_i(u_i, z_0) = \max_{z_0 \in \bar{Z}} \langle \Pi(\alpha) z_0, z_0 \rangle$$

over all admissible locations α (Π is the solution to the algebraic Riccati equation, ARE). As reported in earlier work, this measure works well for a number of problems. However, it doesn't extend to the case of multiple actuators (actuator locations are chosen to be co-located). As a result, we are looking at control performance measures that incorporate robustness. Thus, we consider spatially distributed disturbance functions while setting up the minimization problem. To do this, we consider the α-parameterized closed loop transfer functions

$$T_{zw}(s;\alpha) = C\left(Is - A - B(\alpha)B^*(\alpha)\Pi\right)D \tag{4}$$

where Π is the solution to either the ARE as above or the H^∞ ARE with a given value of the RMS bound γ.

The H^∞-norm of the above transfer function with either value of Π are similar. In Figures 14, we plot a collection of disturbance functions along with the associated design objective functions (H^∞-norm of the transfer function T_{zw} using Π from the ARE). Note that the optimal actuator position would be in the center of the rod without any disturbances. The effect of the disturbances is to shift the minima closer to the region where the disturbance has "more support." We continue to seek appropriate quantification of the best actuator location.

LQR for Index-2 DAEs

Early studies of linear quadratic regulator (LQR) control for differential algebraic equations (DAEs) typically consider singular perturbations and *index 1* problems. The literature on

index 2 systems is much more limited. Our research considered a special form of the singular, linear time-invariant index 2 system

$$E\dot{x}(t) = Ax(t) + Bu(t), \qquad x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \in \mathbb{R}^{n=r+s}, \quad u(t) \in \mathbb{R}^m, \tag{5}$$

for $t > 0$. We assume that the matrices above have the following structure:

$$\begin{aligned} E &= \begin{bmatrix} E_{11} & 0 \\ 0 & 0 \end{bmatrix}, \qquad E_{11} \in \mathbb{R}^{r\times r} \quad \text{with} \quad \text{rank}(E_{11}) = r, \\ A &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & 0 \end{bmatrix}, \qquad A_{11} \in \mathbb{R}^{r\times r}, \quad A_{21} \in \mathbb{R}^{s\times r} \quad \text{and} \quad A_{12} = A_{21}^T, \\ B &= \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \qquad B_1 \in \mathbb{R}^{r\times m}, \quad \text{and} \quad B_2 \in \mathbb{R}^{s\times m}. \end{aligned}$$

For compatibility we require that the columns of B_2 lie in the range of A_{21} (in some cases, B_2 is zero). We also assume, for convenience, that A_{21} has full row rank (otherwise a change of variables can be used to remove redundancies). For many practical problems E and A are symmetric matrices.

For the well-posedness of system (5), we quote the theory of linear autonomous DAEs. Namely, we will assume that the initial conditions for x_1 are consistent, $A_{21}x_1(0) = 0$ (or $A_{21}x_1(0) + B_2u(0) = 0$), that the control is differentiable, and that $sE - A$ is non-singular for a real value of s. This would hold if, e.g. A is invertible.

Systems of this form arise in a number of important problems including discretizations of a class of saddle point problems. An important example for this research is a standard mixed formulation for the Stokes equations. Let the velocity $v \in \mathcal{V}$ and pressure $p \in \mathcal{P}$ be approximated using bases $\{\phi_j\}_{j=1}^r$ and $\{\psi\}_{j=1}^s$. The weak form (leading to finite element equations) of the problem is: Find $\mathbf{v}^n(\cdot,t) \in \mathcal{V}$ and $p^n(\cdot,t) \in \mathcal{P}$ such that for $t > 0$

$$\langle \mathbf{v}_t^n, \phi_i\rangle = -\langle \mu\nabla\mathbf{v}^n, \nabla\phi_i\rangle + \langle p^n, \nabla\cdot\phi_i\rangle + \sum_{j=1}^m \langle \mathbf{b}_j, \phi_i\rangle u_j \qquad \text{for} \quad i = 1,\ldots,r \tag{6}$$

$$0 = \langle \nabla\cdot\mathbf{v}^n, \psi_i\rangle \qquad \text{for} \quad i = 1,\ldots,s. \tag{7}$$

The boundary integral terms all vanish due to the choice of boundary conditions above. This discretization produces a system of the form (5) with $(E_{11})_{ij} = \langle\phi_j,\phi_i\rangle$, $(B_1)_{ij} = \langle \mathbf{b}_j, \phi_i\rangle$,

$$(A_{11})_{ij} = -\mu\langle\nabla\phi_j, \nabla\phi_i\rangle, \quad (A_{12})_{ij} = \langle\psi_j, \nabla\cdot\phi_i\rangle, \quad \text{and} \quad (A_{21})_{ij} = \langle\nabla\cdot\phi_j, \psi_i\rangle.$$

The coefficients of $\mathbf{v}$ correspond to x_1 and the coefficients of p correspond to x_2.

Note that this control problem would result if one were to discretize the linearized Navier-Stokes equations (known as the Oseen equations). The difference would appear in the A_{11} block, which would have an additional term $(-\phi_j\cdot\nabla U - U\cdot\nabla\phi_j, \phi_i)$, where U is the nominal flow that one linearizes about. If one linearizes about zero, the above Stokes equations are obtained. Thus, we may consider large values of μ which would otherwise violate the Stokes hypothesis.

We consider the feedback control for finite dimensional systems in the form (5). Thus, we seek the control $u(t)$ in the form

$$u(t) = -Kx_1(t) \tag{8}$$

that minimizes the cost $J(u, x_{\text{init}}) = \int_0^\infty \{\langle Cx_1, Cx_1\rangle_{\mathbb{R}^r} + \langle u, Ru\rangle_{\mathbb{R}^m}\}\, dt$ subject to the dynamics given in (5).

In the next section we propose four algorithms for solving the feedback control problem. Two of these algorithms use index reduction and are related to well-known methods for simulating Navier-Stokes equations: pseudo-compressibility and a penalty method. The other two are based on generating conforming discretizations (where the constraint equation is automatically satisfied). One of these methods handles the problem as a post-processing step, while the final method requires special discretization and is not always practicable.

Algorithm A.1: The Pseudo-Compressibility Algorithm

The first is to perturb the problem to a system of differential equations by adding a block to the left hand side matrix

$$E = \begin{bmatrix} E_{11} & 0 \\ 0 & -\epsilon M \end{bmatrix}, \tag{9}$$

where M is an easily invertible matrix. For the solution to the Stokes equations, this is referred to as the "pseudo-compressibility" method and was first proposed by Chorin for incompressible Navier-Stokes simulations. (This amounts to a singular perturbation of Stokes flow.) Thus, the control is designed based on the system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} E_{11}^{-1}A_{11} & E_{11}^{-1}A_{21}^T \\ -\frac{1}{\epsilon}M^{-1}A_{21} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} E_{11}^{-1}B_1 \\ -\frac{1}{\epsilon}M^{-1}B_2 \end{bmatrix} u. \tag{10}$$

Since we are only looking at the control in terms of x_1, we find K as the restriction of the gain computed above to x_1.

Algorithm A.2: The Penalty Method Algorithm

The second algorithm is based on imposing the constraint through a penalty method

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & \epsilon M \end{bmatrix}. \tag{11}$$

Again, the matrix M is selected as an easily invertible matrix. This is a computational mechanism for solving Stokes equations, known as the penalty method. Upon inversion of ϵM, we can reformulate problem (5) as a differential equation for x_1.

$$\dot{x}_1 = E_{11}^{-1}\left(A_{11} - \frac{1}{\epsilon}A_{21}^T M^{-1} A_{21}\right) x_1 + E_{11}^{-1}\left(B_1 - \frac{1}{\epsilon}A_{21}^T M^{-1} B_2\right) u. \tag{12}$$

The control for the above equation is naturally formulated in terms of x_1.

Algorithm 2.3: The Projection Algorithm

The third algorithm explicitly represents the problem as a differential equation on a manifold. This is only practically implementable if $B_2 = 0$, so we consider this case first, then state the results for the general case. If V_2 is a collection of s orthonormal column vectors and span(V_2) is the null space of A_{21}, then let $x_1 = V_2 c$, and upon premultiplication by V_2^T, the first equation becomes

$$V_2^T E_{11} V_2 \dot{c} = V_2^T A_{11} V_2 c + V_2^T A_{12} x_2 + V_2^T B_1 u \tag{13}$$

where the term $V_2^T A_{12} x_2$ vanishes since $A_{21} V_2 = 0$. Thus, only a control problem for the new variable c remains,

$$\dot{c} = \left(V_2^T E_{11} V_2\right)^{-1} \left(V_2^T A_{11} V_2\right) c + \left(V_2^T E_{11} V_2\right)^{-1} V_2^T B_1 u.$$

This is a pure linear algebra approach leading to the smallest possible control problem in $r - s$ variables . Note that we must place additional restrictions on V_2 to ensure that the control generated for c is suitable for the original problem.

Finally, if $B_2 \neq 0$, then the constraint equation requires B_2 to be in the range of A_{21}. Thus, we will represent $B_2 = A_{21}\eta$ for some $\eta \in \mathbb{R}^{s \times m}$. Using the specific form of the control (8), we have $A_{21} x_1 - A_{21} \eta K x_1 = 0$. Thus, if we could find a basis V_2 for the matrix $A_{21} - A_{21}\eta K$, then we could carry out the discussion at the beginning of this section. However, we will not generally know K *a priori*, limiting the applicability of this approach.

We consider controlling Stokes equations in a unit box with a rotational body force $\mathbf{b} = (.5 - y, x - .5)$ and a viscous fluid with $\mu = 1000$. Taylor hood elements are used to generate the discretized problems with N elements per side (leading to $r = 2(2N-1)^2$ and $s = (N+1)^2$). We use the results from Algorithm A.3 as the true solution and tabulate the errors using Algorithm A.1 and Algorithm A.2 in Table 8.

Again, we see the advantage of the penalty method over the pseudo-compressibility method. However, in both cases, the convergence is much slower in ϵ than the previous example. The L^2-norm of the error also converges slowly as the mesh is refined. Note that the penalty method approach ultimately broke down for ϵ =1e-10 and $N = 12$ or 14. This suggests that the stability of these ϵ-approximate approaches need to be considered as we take ϵ very small. This is addressed in the section below. We note that the accuracy of these ϵ-approximate algorithms is a function of the viscosity parameter μ. When $\mu = 0.1$, we achieve similar tables, but the errors are approximately four orders of magnitude larger.

As seen above, in comparing these approaches, we must not only compare the accuracy of the approximate solutions, but also the conditioning (sensitivity to first order perturbations) of the resulting algebraic Riccati equations (ARE)

$$AX + A^T X + Q - XBR^{-1}B^T X = 0 \quad \text{where} \quad Q = C^T C.$$

The condition number, denoted by κ, is an indication of the sensitivity of the solution to perturbations in the data.

Algorithm A.1, Pseudo-Compressibility Algorithm				
N	ϵ =1e-4	ϵ =1e-6	ϵ =1e-8	ϵ =1e-10
4	2.715244e-06	2.694859e-06	1.726598e-06	5.827051e-07
6	2.701433e-06	2.679045e-06	1.665645e-06	5.259460e-07
8	2.701604e-06	2.678580e-06	1.650391e-06	5.088347e-07
10	2.707469e-06	2.684163e-06	1.648144e-06	4.953090e-07
12	2.715604e-06	2.692135e-06	1.650872e-06	4.816763e-07
14	2.723676e-06	2.700094e-06	1.654640e-06	4.675749e-07
Algorithm A.2, Penalty Method Algorithm				
4	5.729313e-07	4.740617e-07	4.745786e-07	4.745611e-07
6	4.941872e-07	3.912325e-07	3.919399e-07	3.920233e-07
8	4.768200e-07	3.735520e-07	3.741555e-07	3.692929e-07
10	4.685493e-07	3.580090e-07	3.582774e-07	4.155175e-07
12	4.632434e-07	3.421274e-07	3.424617e-07	*ill-posed*
14	4.593974e-07	3.272885e-07	3.282849e-07	*ill-posed*

Table 8: Stokes Equation: Errors in Functional Gains

We compute bounds on the condition number as suggested by Kenney and Hewer. They extended the ideas of Byers, and obtained a sharper bound for the approximate Byers condition number. They also extended the norms to norms other than the Frobenius norm. If κ_L and κ_B respectively denote the lower and upper bounds defined by Kenney and Hewer, then

$$\frac{\kappa_L}{3} \leq \kappa \leq \kappa_U$$

where the spectral norm is used. The numerical study of condition number growth is provided in Table 9.

Stokes Equation Example				
Alg. A.1	4	$\kappa \in$ (7.10e+3, 6.13e+4)	$\kappa \in$ (2.38e+5, 1.90e+6)	$\kappa \in$ (9.04e+7, 1.13e+9)
Alg. A.2		$\kappa \in$ (7.47e+1, 2.57e+2)	$\kappa \in$ (5.47e+3, 1.83e+5)	$\kappa \in$ (5.46e+5, 1.83e+6)
Alg. A.3		$\kappa \in$ (3.38e+1, 1.10e+2)	$\kappa \in$ (3.38e+1, 1.10e+2)	$\kappa \in$ (3.38e+1, 1.10e+2)
Alg. A.1	8	$\kappa \in$ (7.53e+3, 1.85e+5)	$\kappa \in$ (2.09e+5, 2.80e+6)	$\kappa \in$ (1.24e+8, 2.02e+9)
Alg. A.2		$\kappa \in$ (3.02e+2, 1.05e+3)	$\kappa \in$ (2.26e+4, 7.74e+4)	$\kappa \in$ (2.26e+6, 7.73e+6)
Alg. A.3		$\kappa \in$ (1.35e+2, 4.56e+2)	$\kappa \in$ (1.35e+2, 4.56e+2)	$\kappa \in$ (1.35e+2, 4.56e+2)
Alg. A.1	12	$\kappa \in$ (9.07e+3, 3.93e+5)	$\kappa \in$ (2.11e+5, 3.81e+6)	$\kappa \in$ (1.58e+8, 2.90e+9)
Alg. A.2		$\kappa \in$ (6.81e+2, 2.35e+3)	$\kappa \in$ (5.12e+4, 1.76e+5)	$\kappa \in$ (5.11e+6, 1.76e+7)
Alg. A.3		$\kappa \in$ (3.17e+2, 1.08e+3)	$\kappa \in$ (3.17e+2, 1.08e+3)	$\kappa \in$ (3.17e+2, 1.08e+3)

Table 9: Stokes Equations: Upper and Lower Bounds on κ

Personnel Supported

Faculty Jeff Borggaard (PI)

Post-Docs John-Paul Roop and Lizette Zietsman

Students Grant Boquet, Martin Frame, Peter Hou, Weston Hunter, Karl Mann, Hoan Nguyen, John Singler, Miroslav Stoyanov, Jerri Sayers, Daniel Sutton, James Vance, Eric Vugrin and Kay Vugrin

Dissemination of Research Results

Publications During this project, we have submitted 45 publications. These have all appeared and are listed below.

1. A General Continuous Sensitivity Equation Formulation for Complex Flows (with D. Pelletier and É. Turgeon), in *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Design*, AIAA Paper Number 2000-4732 (September 2000).

2. A Continuous Sensitivity Equation Method for Flows with Temperature Dependent Properties (with D. Pelletier and É. Turgeon), in *Proceedings of the 8th AIAA/ USAF/ NASA/ ISSMO Symposium on Multidisciplinary Analysis and Design*, AIAA Paper Number 2000-4821 (September 2000).

3. Sensitivity Analysis for Chemical Laser Design: A Model Problem (with E. Cliff), in *Proceedings of the 2000 IEEE International Conference on Control Applications*, pages 519–523 (September 2000).

4. On Efficient Solutions to the Continuous Sensitivity Equation Using Automatic Differentiation (with A. Verma), *SIAM Journal on Scientific Computing*, Vol. 22, No. 1, pages 39–62 (2001).

5. Sensitivity and Uncertainty Analysis for Variable Property Flows (with D. Pelletier and É. Turgeon), in *Proceedings of the 39th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper Number 2001-0140 (2001).

6. Adaptivity, Sensitivity and Uncertainty: Towards Standards in CFD (with D. Lacasse, D. Pelletier and É. Turgeon), in *Proceedings of the 39th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper Number 2001-0192 (2001).

7. A Sensitivity Equation Method for Conduction and Phase Change Problems (with D. Pelletier), in *Nonsmooth/Nonconvex Mechanics: Modeling, Analysis and Numerical Methods*, (Nonsmooth Optimization and Applications Series, Volume 45), D. Gao, R. Ogden and G. Stavroulakis, editors, Kluwer Academic, pages 43–68 (2001).

8. Application of a Sensitivity Equation Method to the k–ϵ Model of Turbulence (with É. Turgeon and D. Pelletier), in *Proceedings of the 15th AIAA Computational Fluid Dynamics Conference*, AIAA Paper Number 2001-2534 (2001).

9. A General Continuous Sensitivity Equation Formulation for the k–ϵ Model of Turbulence (with É. Turgeon and D. Pelletier), in *Proceedings of the 15th AIAA Computational Fluid Dynamics Conference*, AIAA Paper Number 2001-3000 (2001).

10. Parametric Uncertainty Analysis for Thermal Fluid Calculations (with D. Pelletier and É. Turgeon), *Journal of Nonlinear Analysis: Series A, Theory and Methods*, Vol. 47, pages 4533–4543 (2001).

11. Reduced Order Controllers for Burgers' Equation with a Nonlinear Observer (with J. Atwell and B.B. King), *International Journal of Applied Mathematics and Computational Science*, Vol. 11, No. 6, pages 1311–1330 (2001).

12. Sensitivity Analysis with Sliding Boundary Conditions (with E. Cliff and A. Godfrey), in *Proceedings of the 40th AIAA Aerospace Sciences Meeting and Exhibition*, AIAA Paper Number 2002-0100 (2002).

13. Sensitivity and Uncertainty Analysis for Turbulent Flows (with S. Étienne, D. Pelletier and É. Turgeon), in *Proceedings of the 40th AIAA Aerospace Sciences Meeting and Exhibition*, AIAA Paper Number 2002-0985 (2002).

14. Reliable Sensitivity Analysis via an Adaptive Sensitivity Equation Method (with D. Pelletier, É. Turgeon and S. Étienne), in *Proceedings of the 3rd Theoretical Fluid Mechanics Meeting*, AIAA Paper Number 2002-2758 (2002).

15. A Continuous Control Design Method (with J. Burns), in *Proceedings of the 3rd Theoretical Fluid Mechanics Meeting*, AIAA Paper Number 2002-2989 (2002).

16. On Sensitivity Analysis for Problems with Numerical Noise (with D. Pelletier and K. Vugrin), in *Proceedings of the 9th AIAA Multidisciplinary Analysis and Optimization Meeting*, AIAA Paper Number 2002-5553 (2002).

17. Parametric Uncertainty Analysis in a Phase Change Model (with J.-F. Hètu and D. Pelletier), in *Proceedings of the 9th AIAA Multidisciplinary Analysis and Optimization Meeting*, AIAA Paper Number 2002-5601 (2002).

18. A General Continuous Sensitivity Equation Formulation for Complex Flows (with É. Turgeon and D. Pelletier), *Numerical Heat Transfer: Part B, Fundamentals*, Vol. 42, No. 6, pages 485–498 (2002).

19. Second-Order Sensitivity Analysis for Conjugate Phase-Change Problems (with D. Pelletier and C. Winter), in *Proceedings of the 41st Aerospace Sciences Meeting and Exhibition*, AIAA Paper Number 2003-0512 (2003).

20. A Second Order Sensitivity Equation Method for Laminar Flows (with S. Étienne, J.-N. Mahieu and D. Pelletier), in *Proceedings of the 10th Annual Conference of the CFD Society of Canada*, (2003).

21. Sensitivity Analysis of Transient Non-Linear Heat Conduction (with S. Étienne, H. Hristova, J.-N. Mahieu and D. Pelletier), in *Proceedings of the 10th Annual Conference of the CFD Society of Canada*, (2003).

22. A Sensitivity Equation Method for Turbulent Heat Transfer (with E. Colin, S. Étienne and D. Pelletier), in *Proceedings of the 36th AIAA Thermophysics Conference*, AIAA Paper Number 2003-3636 (2003).

23. Optimization of an Integrated Actuator Placement and Robust Control Scheme for Distributed Parameter Processes Subject to Worst-Case Spatial Disturbance Distribution (with M. Demetriou), in *Proceedings of the 2003 American Control Conference*, (2003).

24. A General Continuous Sensitivity Equation Formulation for the $k - \epsilon$ Model of Turbulence (with É. Turgeon and D. Pelletier), *International Journal of Computational Fluid Dynamics*, Vol. 18, No. 1, pages 29–46 (2004).

25. Application of a Sensitivity Equation Method to Transient Non-Linear Heat Conduction (with H. Hristova, S. Etienne and D. Pelletier), in *Proceedings of the 42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper Number 2004-0495 (2004).

26. Second Order Sensitivity and Uncertainty Analysis of Laminar Airfoil Flows (with J. Mahieu, D. Pelletier and J. Trepanier), in *Proceedings of the 42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper Number 2004-0742 (2004).

27. Application of a Sensitivity Equation Method to Turbulent Flows with Heat Transfer (with E. Colin, S. Etienne and D. Pelletier), in *Proceedings of the 42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper Number 2004-1290 (2004).

28. Sensitivity Equations for the Design of Control Systems (with J. Vance), in *Proceedings of the Sixth IASTED International Conference on Control and Applications*, IASTED Paper Number 441-050 (2004).

29. Computational Challenges in Control of Partial Differential Equations (with J. Burns and L. Zietsman), in *Proceedings of the 2nd AIAA Flow Control Conference*, AIAA Paper Number 2004-2526 (2004).

30. Optimization of a joint sensor placement and robust estimation scheme for distributed parameter processes subject to worst case spatial disturbance distributions (with M. Demetriou), in *Proceedings of the 2004 American Control Conference*, (2004).

31. A Continuous Sensitivity Equation Method for Time-Dependent Incompressible Laminar Flows (with H. Hristova, S. Etienne and D. Pelletier), in *Proceedings of the 34th AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper Number 2004-2630 (2004).

32. On Strong Convergence of Feedback Operators for Non-Normal Distributed Parameter Systems (with J. Burns, E. Vugrin and L. Zietsman), in *Proceedings of the 43rd*

IEEE Conference on Decision and Control, IEEE Paper Number WeA04.5, pages 1526–1531, (2004).

33. Simulating the Fate of Subsurface-Banded Urea (with S.B. Shah and M.L. Wolfe), *Nutrient Cycling in Agroecosystems*, Vol. 70, pages 47–66 (2004).

34. A Second-order Sensitivity Equation Method for Laminar Flows (with S. Étienne, J.-N. Mahieu and D. Pelletier), *International Journal of Computational Fluid Dynamics*, Vol. 19, No. 2, pages 143–157 (2005).

35. Application of a Sensitivity Equation Method to Turbulent Conjugate Heat Transfer (with E. Colin, S. Etienne and D. Pelletier), in *Proceedings of the 43rd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper Number 2005-0186 (2005).

36. Design of Worst Spatial Distribution of Disturbances for a Class of Parabolic Partial Differential Equations (with M. Demetriou), in *Proceedings of the 2005 American Control Conference*, (2005).

37. Optimal Shape Design in Mixed Convection using a Continuous Sensitivity Equation Approach (with R. Duvigneau and D. Pelletier), in *Proceedings of the 38th AIAA Thermophysics Conference*, AIAA Paper Number 2005-4823 (2005).

38. A Sensitivity Equation Method for Compressible Subsonic Laminar Airfoil Flows (with P. Edmond, D. Pelletier and S. Etienne), in *Proceedings of the 23rd AIAA Applied Aerodynamics Conference*, AIAA Paper Number 2005-4601 (2005).

39. A Continuous Second Order Sensitivity Equation Method for Time-Dependent Incompressible Laminar Flows (with F. Ilinca and D. Pelletier), in *Proceedings of the 17th AIAA Computational Fluid Dynamics Conference*, AIAA Paper Number 2005-5252 (2005).

40. Application of a Sensitivity Equation Method to Turbulent Flows with Heat Transfer (with E. Colin, S. Étienne and D. Pelletier), *International Journal of Thermal Sciences*, Vol. 44, No. 11, pages 1024–1038 (2005).

41. Application of a Sensitivity Equation Method to Compressible Subsonic Impinging Jets (with P. Edmond, D. Pelletier, S. Etienne, and A. Hay), in *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper Number 2006-0909 (2006).

42. A Continuous Sensitivity Equation Method for Time-dependent Incompressible Laminar Flows (with H. Hristova, S. Étienne, and D. Pelletier), *International Journal for Numerical Methods in Fluids*, Vol. 50, No. 7, pages 871–844 (2006).

43. A General Sensitivity Equation Formulation for Turbulent Heat Transfer (with E. Colin, S. Étienne and D. Pelletier), *Numerical Heat Transfer: Part B, Fundamentals*, Vol. 49, No. 2, pages 125–153 (2006).

44. Approximate Deconvolution Boundary Conditions for Large Eddy Simulation (with T. Iliescu), *Applied Math Letters*, Vol. 19, pages 735–740 (2006).

45. An Improved Continuous Sensitivity Equation Method for Optimal Shape Design in Mixed Convection (with R. Duvigneau and D. Pelletier), *Numerical Heat Transfer: Part B, Fundamentals*, Vol. 50, No. 1, pages 1–24 (2006).

Presentations at meetings, conferences or seminars

During this project, we have given over 65 presentations at meetings, conferences or seminars. Presentations given by the PI are listed below.

1. CFD2K, 8th Annual Conference of the CFD Society of Canada, Montréal, Québec (June 2000).

2. 2000 American Control Conference, Chicago, IL (June 2000).

3. 3rd World Congress on Nonlinear Analysis, Catania, Italy (July 2000).

4. United Technologies Research Center, East Hartford, CT (July 2000).

5. AFOSR Workshop on Dynamics and Control, Pasadena, CA (August 2000).

6. 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA (September 2000).

7. Boeing Seminar on Control and Design, Boeing Aerospace, Seattle, WA (September 2000).

8. United Technologies Research Center, Project Summary, East Hartford, CT (December 2000).

9. Sandia National Labs, CSRI Seminar Series, Livermore, CA (March 2001).

10. Virginia Tech, Aerospace and Ocean Engineering Seminar, Blacksburg, VA (April 2001).

11. 31st AIAA Fluid Dynamics Conference and Exhibit, Anaheim, CA (June 2001).

12. SIAM Control Conference, San Diego, CA (July 2001).

13. AFOSR Workshop on Dynamics and Control, Dayton, OH (July 2001).

14. Sensitivity Analysis Workshop 2001, Livermore, CA (August 2001).

15. Center for Turbomachinery and Compressor Design Annual Review, Blacksburg, VA (September 2001).

16. University of Trier, Numerical Analysis Seminar, Trier, Germany (November 2001).

17. Iowa State University, Mathematics Colloquium, Ames, IA (March 2002).

18. 3rd AIAA Theoretical Fluids Meeting, 1st AIAA Flow Control Meeting, St. Louis, MO (June 2002). (1 hour invited lecture)

19. SIAM Annual Meeting, Philadelphia, PA (July 2002).

20. Fifteenth International Symposium on Mathematical Theory of Networks and Systems, South Bend, IN (August 2002).

21. AFOSR Workshop on Dynamics and Control, Pasadena, CA (August 2002).

22. 9th AIAA Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA (September 2002).

23. 9th AIAA Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA (September 2002).

24. Worcester Polytechnic Institute, Mechanical Engineering Colloquium, Worcester, MA (September 2002).

25. 22nd Annual Southeastern-Atlantic Regional Conference on Differential Equations, Knoxville, TN (October 2002).

26. IMA Workshop on Optimization in Simulation-Based Models, Minneapolis, MN, (January 2003).

27. Virginia Tech, Mathematics Colloquium, Blacksburg, VA (January 2003).

28. Montana State University, Applied Mathematics Seminar, Bozeman, MT (February 2003).

29. SIAM Conference on Computational Science and Engineering, San Diego, CA (February 2003).

30. University of Louisville, Mathematics Colloquium, Louisville, KY (February 2003).

31. 27th Annual Conference of the South African Society for Numerical and Applied Mathematics, Stellenbosch, South Africa (March 2003).

32. First Joint CAIMS/SIAM Annual Meeting, Montréal, Québec (June 2003).

33. SciCADE 2003, International Conference on Scientific Computation and Differential Equations, Trondheim, Norway (June 2003).

34. 36th AIAA Thermophysics Conference, Orlando, FL (June 2003).

35. 7th US National Congress on Computational Mechanics, Albuquerque, NM (July 2003).

36. United States Air Force Academy, Seminar in Closed Loop Flow Control, Colorado Springs, CO (July 2003).

37. Computation, Control and Biological Systems VIII, Bozeman, MT (July 2003).

38. Wright-Patterson Air Force Base, National Research Council Summer Faculty Seminar, Wright-Patterson Air Force Base, OH (August 2003).

39. AFOSR Workshop on Dynamics and Control, Destin, FL (September 2003).

40. Ohio State University, Collaborative Center for Control Science, Columbus, OH (September 2003).

41. AMS 2003 Fall Southeastern Sectional Meeting, Chapel Hill, NC (October 2003).

42. 23rd Annual Southeastern-Atlantic Regional Conference on Differential Equations, Atlanta, GA (October 2003).

43. Florida State University, School of Computational Science and Information Technology, Tallahassee, FL (November 2003).

44. George Mason University, Mathematics Colloquium, Alexandria, VA (November 2003).

45. SIAM Parallel Processing for Scientific Computing, San Francisco, CA (February 2004).

46. CSIT Workshop on Emerging Methods for Numerical Solution of PDEs, Tallahassee, FL (March 2004).

47. University of Florida Graduate Education Research Center, Seminar, Destin, FL (March 2004).

48. 2004 Advanced Simulation Technologies Conference, Alexandria, VA (April 2004).

49. Optimization Days, Montreal, Canada (May 2004).

50. IFIP Workshop on Shape Optimization and Control, Lisbon, Portugal (June 2004).

51. 2nd AIAA Flow Control Conference, Portland, OR (June 2004).

52. SIAM Annual Meeting, Portland, OR (July 2004).

53. 10th AIAA Symposium on Multidisciplinary Analysis and Optimization, Albany, NY (September 2004).

54. Argonne National Laboratories, Wilkinson Visitor Program, Argonne, IL (October 2004).

55. AMS 2004 Fall Southeastern Sectional Meeting, Pittsburgh, PA (November 2004).

56. American Physical Society, 57th Annual Meeting of the Division of Fluid Dynamics, Seattle, WA (November 2004).

57. IFIP Workshop on Free and Moving Boundaries, Analysis, Simulation and Control, Houston, TX, (December 2004).

58. George Mason University, Fairfax, VA (February 2005).

59. International Conference on Approximation Methods for Design and Control, Buenos Aires, Argentina (March 2005).

60. American Control Conference, Portland, OR (June 2005).

61. SIAM Annual Meeting, New Orleans, LA (July 2005).

62. MOPTA 05, Modeling and Optimization: Theory and Applications, Windsor, ON, Canada, (July 2005). (1 hour invited lecture)

63. AFOSR Workshop on Computational Mathematics, Long Beach, CA (August 2005).

64. Workshop on Large-Scale Robust Optimization, Sante Fe, NM (September 2005).

65. Austrian Mathematical Society, Klagenfurt, Austria (September 2005).

Interactions and Transitions

AeroSoft, Blacksburg, VA (several meetings during the grant period)

Along with Andy Godfrey (AeroSoft) and Gene Cliff (Virginia Tech): Discussed strategies to solve the problem of accurate sensitivity calculations for turbulent flows with geometric parameters.

Along with Andy Godfrey (AeroSoft) and Gene Cliff, Uri Vandsburger (Virginia Tech): Sensitivity analysis played an important role in a combustor design problem investigated as part of an STTR at AeroSoft. Two issues that were raised were (i.) developing appropriate boundary conditions specific to the sensitivity of injector locations and (ii.) demonstrating sensitivity analysis for a nonlinear time-dependent problem.

[Contact: Andy Godfrey (540) 557-1907]

Air Force Research Laboratory, Wright-Patterson Air Force Base, OH

The PI spent the summer of 2003 as a research faculty fellow at the Air Vehicles Directorate working with Siva Banda, Chris Camphouse and James Myatt. While at AFRL, the work on Chandrasekhar integrator for fast functional gain calculations was carried out. This approach resulted in up to an order of magnitude speedup in computational performance. We also outlined several strategies for computing functional gains for Stokes equations.

Daniel Sutton, Masters student in Mechanical Engineering spent the summer at the Air Vehicles Directorate working with Siva Banda, Chris Camphouse and James Myatt: Worked on computing low-order models based on Proper Orthogonal Decomposition. He is looking into POD for coupled systems. The PI and Lizette Zietsman made a two day visit to the lab during Daniel's internship to discuss his research.

[Contacts: Chris Camphouse (937) 255-6326, James Myatt (937) 255-8498]

Air Force Research Laboratory, Eglin Air Force Base, FL

The PI and Ekkehard Sachs (Virginia Tech) met with Major Bill Hilbun (7 Dec. 2003, 11 March 2004) to discuss the modeling of plasma actuators.

Industrial Materials Institute, Boucherville, Québec (June 13, 2000)

Met with Jean-Francois Hétu (IMI) and Dominique Pelletier (École Polytechnique de Montréal): The sensitivity of the location of mold filling lines is a direct application of the research on sensitivity analysis with sliding boundary conditions. We jointly developed models for sensitivity analysis for phase change problems arising in die casting and a contact resistance model (to model the interface between the mold and the part). This research was incorporated in the IMI mold-filling software simulator.

[Contact: Jean-Francois Hétu (450) 641-5000 x35082]

United Technologies Research Center, East Hartford, CT (September / December, 2000)

Met with Andy Godfrey (AeroSoft), Joel Wagner and Brent Staubach (Pratt and Whitney), John Whiton (International Fuel Cell) and Mike Dorobantu, Bob LaBarre and David Sirag (UTRC): Discussed two model problems to demonstrate the feasibility of using a sensitivity solver as a post-processing module to either a commercial CFD solver such as Fluent or existing in-house CFD solvers.

These problems stressed current capabilities of AeroSoft's SENSE package which does not support sensitivity analysis for geometric variables in turbulent flows. Another limitation is that some models used in Fluent are black-box and appropriate sensitivity equations for these unknown models can't be developed without information about these models.

[Contact: Mike Dorobantu (860) 610-7824]

Honors/Awards

Air Force Presidential Early Career Award for Scientists and Engineers (April 2000).
National Research Council Summer Faculty Fellowship (May-August 2003).
Virginia Tech Mathematics Professor of the Year (2004).